

# Interaktion

July 1985

IUGN-8

## CONTENTS

	Page
EDITORIAL .....	2
CUSTOM 80 MODIFICATION .....	2
CUSTOM 80 PROGRAM .....	3
CP/M ON THE INTERAK (PART 3) .....	5
LETTERS .....	19
PRINT DECIMAL OF HL REGISTER PAIR .....	22
BOOK LIBRARY LISTING .....	23
SOFTWARE LIBRARY LISTINGS .....	24
CONTACTS .....	25

All published items remain the copyright of the originators. Members may use published items for their own enjoyment and education but must not describe as their own work or offer for sale any item or part of any item published herein without the express permission of the originator.

## Editorial

Dear Member,

Thanks to all of the members who have submitted items for publication. As you probably realise it is impossible to reply directly to all who write to me, but all submitted articles will be published in the near future.

ZYBASIC has been slightly updated. Nothing spectacular, just a few bugs fixed. VDU-2K version is now official. Contact Greenbank for details of how to upgrade.

Mr D.Mason is willing to run an Interak Bulletin Board. He can also provide software to drive the application. See Contacts page for his address.

Bob Eldridge

-----  
CUSTOM 80 MODIFICATION

BY DON COX

## Problem:

Intermittent failure of the printer.  
It appeared to be receiving one or two wrong bits in character.

## Solution:

Investigation revealed that the frequency of the master clock oscillator on the cassette interface board was varying, especially when the computer was cold. The most likely cause of this was the use of a ceramic capacitor for C10; according to RS data sheet 2113, dealing with 555 circuits, ceramic capacitors are not stable in this type of circuit. It is also quite difficult to adjust VR1 correctly.

Therefore C10 was replaced by a polystyrene capacitor of 1000pF (=1nF) and VR1 was replaced by a multiturn Cermet resistor mounted on the front panel. A suitable mounting bracket is available from RS Components. Note that only two wires have to be connected to the Cermet. At the same time, link W1 was replaced by a switch mounted on the front panel; this allows the clock frequency to be conveniently monitored at any time. It is important to keep the Cermet and switch over to the left so that they do not foul the RAM board in the next socket.

## Result:

The clock frequency is now extremely stable.

If a 10K Cermet is used, a wider range of adjustment is possible, and Baud rates other than the standard four can be used. In principle, one could load and make listings of programs written for Spectrums, for instance (1500 Baud), but I haven't tried this.

Note that if you aren't running the DFM program, the CTC is doing nothing.

CUSTOM 80 PROGRAM  
BY DON COX

;TICKER

ORG 2000H  
LOAD 2000H

KBINVT: EQU 0EH  
CURSOR: EQU 0FFECH  
STATUS: EQU 0FFE1H  
CURCHR: EQU 0FFE2H

```

2000 2A0E00      LD HL,(KBINVT)
2003 220E21      LD (VECTORS+KBINVT),HL
2006 210021      LD HL,VECTORS ;START OF TABLE OF ADDRESSES
2009 7C          LD A,H
200A ED47        LD I,A ;HIGH BYTE GOES TO VECTOR
                    ;...REGISTER

200C ED5E        IM 2
200E 7D          LD A,L
200F C610        ADD A,10H ;VECTORS FOR CTC CAN START ABOVE
                    ;...THE KEYBOARD VECTOR
2011 D304        OUT (04),A ;TO CTC 0 FOR LOW BYTE OF FIRST
                    ;...VECTOR
2013 3E03        LD A,03 ;RESET NUMBER
2015 D304        OUT (04),A ;SEND TO CHANNELS 0 & 1 OF CTC
2017 D305        OUT (05),A
2019 3E25        LD A,25H ;CONTROL WORD
201B D304        OUT (04),A
201D 3E7D        LD A,125 ;TIME CONSTANT
201F D304        OUT (04),A
2021 3EC5        LD A,0C5H ;CONTROL WORD
2023 D305        OUT (05),A
2025 3E3F        LD A,63
2027 D305        OUT (05),A
2029 FB          EI
202A C32D20      JP TYPE ;OR WHEREVER
;THIS COULD ALSO BE A RST 28H JP 0 OR RESET STOPS THE INTERRUPT

```

;THIS IS AN EXAMPLE OF A PROGRAM THAT CAN RUN ALONG WITH INTERRUPTS

```

202D DF          TYPE: RST 18H
202E 03          DB 03 ;FETCH CHARACTER FROM KEYBOARD
202F DF          RST 18H
2030 05          DB 05 ;DISPLAY IT
2031 18FA        JR TYPE

```

```

2033 F5          INTERRUPT:PUSH AF
2034 E5          PUSH HL

```

;CURSOR POSITIONS HAVE TO BE STORED IF THE INTERRUPT  
;...INVOLVES SCREEN DISPLAY

```

2035 2AECFF      LD HL,(CURSOR)
2038 229320      LD (CURSTORE),HL
203B 2A9520      LD HL,(CURSTORE2)
203E 22ECFF      LD (CURSOR),HL

```

```

2041 3AE2FF          LD A,(CURCHR)
2044 329720          LD (CHARSTR),A
2047 3AE1FF          LD A,(STATUS)
204A 329222          LD (STATUSTR),A
204D CF             RST 08
204E 1D00           DB 1D,00          ;SET SCREEN TO PAGE MODE
2050 3A9820          LD A,(LATCH)
2053 CB47           BIT 0,A
2055 280C           JR Z,TOCK

2057 CF             TICK:  RST 08
2058 20205449        DB ' TICK ',00
205C 434B2020
2060 00
2061 180A           JR EXIT

2063 CF             TOCK:  RST 08
2064 2020544F        DB ' TOCK ',00
2068 434B2020
206C 00
206D 3A9820          EXIT:  LD A,(LATCH)
2070 EE01           XOR 01
2072 329820          LD (LATCH),A
2075 3A9220          LD A,(STATUSTR)
2078 32E1FF          LD (STATUS),A
                ;THIS SETS SCREEN BACK TO SCROLL MODE, IF ON
207B 3A9720          LD A,(CHARSTR)
207E 32E2FF          LD (CURCHR),A
2081 2AECFF          LD HL,(CURSOR)
2084 229520          LD (CURSTORE2),HL
2087 2A9320          LD HL,(CURSTORE)
208A 22ECFF          LD (CURSOR),HL
208D E1             POP HL
208E F1             POP AF
208F FB             EI
2090 ED4D           RETI
2092 00             STATUSTR: DB 0
2093 0000            CURSTORE: DW 0
2095 0000            CURSTORE2: DW 0
2097 00             CHARSTR:  DB 00
2098 00             LATCH:    DB 00

                ORG 2100H
                LOAD 2100H
2100 0000            VECTORS:  DW 0
2102 0000            DW 0
2104 0000            DW 0
2106 0000            DW 0
2108 0000            DW 0
210A 0000            DW 0
210C 0000            DW 0
210E 0000            DW 0 ;KEYBOARD INTERRUPT VECTOR COPIED TO HERE
2110 0000            DW 0
2112 0000            DW INTERRUPT
                END

```



CP/M ON THE INTERAK (PART 3)  
 INTERAK BOOT ROM 0301-0100-018401  
 BY R.ELDRIDGE

This boot rom is a first write for the Interak using a VDU 2K. It follows the "BOOT LAW" in that a jump table is the link to peripheral dependant subroutines. In all boot ROMs the jump table remains the same, only the code of the routines are different. This has the effect of allowing any Interak disk to boot up and run on any Interak, even if one machine uses a VDU-2K and the other uses a serial terminal. Each will have the correct boot ROM and translation will be automatic. Write your BIOS to link to the jump table and it will run on all Interak's.

This Boot Rom will boot from either single or double density 8 inch disks. Self selected.

Single density is 77 tracks of 26 sectors of 128 bytes.

Double density is 77 tracks of 26 sectors of 256 bytes.

```
; BOOT VVKK-PPDD-NNYYMM
;   VV.....Vdu code
;   KK.....Keyboard code
;   PP.....Printer code
;   DD.....Disk drive code
;   NN.....Version number
;   YY.....Year
;   MM.....Month
```

```
;Version numbers used so far
;01.....First build.
```

```
;Vdu codes (known at this time)
;01.....32 * 24, memory mapped. 0F000H. Vdu K.
;02.....64 * 16, memory mapped. 0F000H. Vdu K+.
;03.....64 * 24, memory mapped. 0F000H. Vdu 2K.
;04.....Intelgraph port mapped at E0, E1
;05.....Intelgraph RS232 on ports 0,1 (stat,dat).
```

```
;Keyboard codes (known at this time)
;01.....8 bit parallel, positive going strobe on
;         bit 7. Via port 40H. LK1.
;02.....RS232 on ports 0,1 (stat,dat).
```

```
;Printer codes (known at this time)
;01.....8 bit parallel ascii data on port 7,
;         Controlled by TBMT on bit 7, port 6.
;02.....RS232 device, stat 02, data 03. Tbmt is bit 7.
```

```
;Disk drive codes (known at this time)
;00.....8 inch single sided single or double density
;         automatic density boot up.
```

```

E000          ORG 0E000H      ;Boot rom start address
BDFF          STACK EQU 0BDFFH ;Stack

;Spad
0040          CURSOR EQU 40H    ;Cursor
0042          KEYCODE EQU CURSOR+2 ;Last key pressed
0043          CURCAR EQU KEYCODE+1 ;Character under cursor
0044          DISKNO EQU CURCAR+1
0045          TRACK EQU DISKNO+1
0046          SECTOR EQU TRACK+1
0047          COMMND EQU SECTOR+1
0048          RXSTAT EQU COMMND+1
0049          RTOP EQU RXSTAT+1 ;Store top

;Ascii keys.
0020          SPACE EQU 20H     ;Ascii space
0008          BS EQU 8         ;Ascii backspace
000D          CR EQU 0DH       ;Ascii carriage return
0003          CTRL_C EQU 3     ;Ascii control and C
0002          CTRL_B EQU 2     ;Ascii control and B
000C          FF EQU 0CH       ;Ascii form feed
007F          DEL EQU 7FH      ;Ascii delete
000A          LF EQU 0AH       ;Ascii line feed

;Vdu 2K parameters.
F000          VDU EQU 0F000H
0040          COLS EQU 64
0018          LINES EQU 24
F5C0          LINE24 EQU 0F5C0H

;LK1 parameters
0040          KDATA EQU 40H

;Disk ports.
0080          SDATA EQU 80H     ;Contains ST-512
0080          DSTAT EQU SDATA  ;Disk status (read)
0080          DCOMM EQU SDATA  ;Disk command (write)
0081          DTRAC EQU DCOMM+1 ;Disk track (read/write)
0082          DSECT EQU DTRAC+1 ;Disk sector (read/write)
0083          DDATA EQU DSECT+1 ;Disk data (read/write)
0084          DPOLL EQU DDATA+1 ;Disk polling (read)
0085          DCONF EQU DPOLL+1 ;Disk config (read/write)
0086          DOPTN EQU DCONF+1 ;Disk options (read)

;Disk commands
001D          SEEK EQU 01DH     ;Seek command
00AC          WRITS EQU 0ACH     ;Write sector command
008C          READS EQU 08CH     ;Read sector command
000D          REST0 EQU 0DH      ;Restore command
0000          RDSEC EQU 0       ;Read sector routine

E000 000000 BEGIN: DEFB 0,0,0 ;User branch out
E003 C370E0          JP COLDST  ;Cold start
E006 C355E0          JP TRAPET  ;Trap error via 0038H
E009 C34BE0          JP BOOTER  ;Warm boot error

```

```

E00C C389E4      JP CONST      ;Get key status
E00F C3A3E4      JP CONIN      ;Get key
E012 C3B6E4      JP CONOUT     ;Print C reg to vdu
E015 C30FE5      JP LIST       ;Print A reg to printer
E018 C369E4      JP LISTST     ;Get printer status
E01B C372E4      JP PUNCH      ;Punch
E01E C36DE4      JP READER     ;Reader
E021 C372E1      JP PRNMSG     ;Print message
E024 C314E2      JP ATODEC     ;Print decimal of A
E027 C347E2      JP HEXACC     ;Print hex of Acc
E02A C33CE2      JP HEXHL      ;Print hex of HL
E02D C37CE1      JP COMPAR     ;Compare HL:DE
E030 C37BE4      JP CVDU       ;Clear the vdu
E033 C365E1      JP DISCOM     ;Send command in A to disk
E036 C381E1      JP DISKER     ;Disk error handler
E039 C350E0      JP LOADER     ;Loader error
E03C C370E1      JP SPARE1     ; *
E03F C371E1      JP SPARE2     ; *
E042 C380E1      JP SPARE3     ; *
E045 C313E2      JP SPARE4     ; *
E048 C338E2      JP SPARE5     ; *

```

```

E04B 210FE3      BOOTER: LD HL,SWBER ;Warm boot error entry
E04E 180D        JR ABORT      ;To retry
E050 2123E3      LOADER: LD HL,SLDER ;Load error entry
E053 1808        JR ABORT      ;To retry
E055 2136E3      TRAPET: LD HL,STRAP ;Trap entry
E058 1803        JR ABORT      ;To retry
E05A 2180E3      CBOOER: LD HL,SCBER ;Cold boot error entry
E05D 31FFBD      ABORT: LD SP,STACK ;Reinitialise the stack
E060 CD72E1      CALL PRNMSG    ;Print boot error message
E063 218FE3      LD HL,SRETRY   ;String to retry
E066 CD72E1      CALL PRNMSG    ;Ask to retry
E069 CDA3E4      WRETRY: CALL CONIN ;Get response
E06C FE0D        CP CR         ;Is response CR?
E06E 20F9        JR NZ,WRETRY   ;If no loop till it is

```

```

;Power on entry.....Initialisation.
E070 DBFF      COLDST: IN A,(0FFH) ;Switch off power jump
E072 2100F0      LD HL,VDU        ;Clear the screen
E075 1101F0      LD DE,VDU+1
E078 010006      LD BC,24*64
E07B 3620        LD (HL),20H
E07D EDB0        LDIR

```

```

;Find a stack parameter. Store top.
E07F 21FFFF      LD HL,0FFFFH    ;Low store
E082 3E55        LD A,55H        ;Write pattern
E084 23          COLDSP: INC HL   ;Next loc
E085 46          LD B,(HL)       ;Save data
E086 77          LD (HL),A       ;Write pattern
E087 BE          CP (HL)         ;Write ok?
E088 70          LD (HL),B       ;Replace data
E089 28F9        JR Z,COLDSP     ;Loop till ram+1
E08B 2B          DEC HL         ;HL points to ram top

```

```

E08C 224900      LD (RTOP),HL      ;Save store top
E08F 25          DEC H            ;ST-256
E090 25          DEC H            ;ST-256
E091 228000      LD (SDATA),HL    ;Save ST-512
E094 F9          LD SP,HL         ;Initialise the stack

                                ;Clear spad
E095 0605        LD B,5           ;Spad entries
E097 214400      LD HL,44H        ;Spad begins
E09A 3E00        LD A,0           ;Clear spad data
E09C 77          CSPAD: LD (HL),A  ;Clear first/next spad
E09D 23          INC HL           ;Next spad
E09E 10FC        DJNZ CSPAD       ;Loop till done
E0A0 3E01        LD A,1           ;Sector initialisation data
E0A2 324600      LD (SECTOR),A    ;Stat sector to 1

                                ;Initialise trap stop.
E0A5 3EC3        LD A,0C3H        ;Opcode of JP instruction
E0A7 323800      LD (38H),A       ;Set JP at 0038H
E0AA 2106E0      LD HL,BEGIN+6    ;Get trap return address
E0AD 223900      LD (39H),HL      ;Form JP trap return at 0038H

                                ;Initialise the FDCI card.
E0B0 3E0C        LD A,0CH         ;Set Acc for ULK,SSSD,8,A.
E0B2 D3B5        OUT (DCONF),A    ;Configure FDCI chip

                                ;Initialise the keyboard.
E0B4 CD74E4      CALL KINIT

                                ;Basic initialisation completed, sign on.
E0B7 211AE5      READY: LD HL,9B0 ;Message to sign on with
E0BA CD72E1      CALL PRNMSG       ;Print sign on message
                                ;Print store size
E0BD 217AE2      LD HL,SMEM        ;Store size message
E0C0 CD72E1      CALL PRNMSG       ;Print store top
E0C3 2A4900      LD HL,(RTOP)      ;Get store top number
E0C6 CD3CE2      CALL HEXHL        ;Print it as hex
E0C9 2156E4      LD HL,STERM       ;Terminate
E0CC CD72E1      CALL PRNMSG       ;Terminate the line
E0CF 11FFBF      LD DE,0BFFFFH     ;DE = 48k top address
E0D2 2A4900      LD HL,(RTOP)      ;HL = Actual store top
E0D5 CD7CE1      CALL COMPAR       ;Compare req to actual.
E0D8 3007        JR NC,RAMOK       ;48k online try to boot
E0DA 21A7E2      LD HL,NSTORE      ;Print "not enough store"
E0DD CD72E1      CALL PRNMSG       ;Print it
E0E0 76          HALT              ;Freeze out.

                                ;Ensure that the drive is ready to Boot.
                                ;If its not then ask the user to ready it and wait
                                ;untill it is ready.
E0E1 2191E2      RAMOK: LD HL,SCHKA ;Checking drive A ready
E0E4 CD72E1      CALL PRNMSG       ;Print it
E0E7 CD60E2      CALL INITRD       ;Ready reader
E0EA 3ED0        FDCRES: LD A,0D0H ;command a reset
E0EC D3B0        OUT (DCOMM),A     ;Execute the reset

```

```

E0EE 0600      LD B,0      ;Delay for reset
E0F0 10FE      WRES:     DJNZ WRES      ;Wait for reset end
E0F2 DB80      IN A,(DSTAT) ;Get drive status
E0F4 CB47      BIT 0,A      ;Test busy bit
E0F6 20F2      JR NZ,FDCRES ;If busy force another reset
E0F8 3ED4      NR:       LD A,0D4H      ;Interrupt at index
E0FA CD65E1    CALL DISCOM ;Command wait for index
E0FD DB80      IN A,(DSTAT) ;Get drive status
E0FF CB7F      BIT 7,A      ;Test for ready
E101 20F5      JR NZ,NR     ;Loop if not ready

;Sign on complete.....Seek track 0 on drive A
E103 060A      LD B,10      ;Restore retry counts
E105 3E0D      QAT00:    LD A,REST0     ;Set Acc to restore command
E107 CD65E1    CALL DISCOM ;Command drive A to restore
E10A DB80      IN A,(DSTAT) ;Get status
E10C CB57      BIT 2,A      ;At track 0?
E10E 2012      JR NZ,AT00   ;Yes. then jump on
E110 10F3      DJNZ QAT00   ;No. then retry if retries
;Seek 0 fails.....Hardware fault!
E112 3E10      LD A,10H     ;For seek error
E114 324800    LD (RXSTAT),A ;Status
E117 3E1D      LD A,SEEK    ;Restore error is seek error
E119 324700    LD (COMMND),A ;Stat command
E11C CD81E1    NBOOT:     CALL DISKER   ;Use disk error handler
E11F C35AE0    JP CBOODER ;Then branch to cboot error

;Get ready to boot.
E122 060A      AT00:     LD B,10      ;Boot retry counter
E124 3E8C      LD A,READS   ;Read sector command
E126 324700    LD (COMMND),A ;Save command
E129 0E84      LD C,DPLL    ;Data port register
E12B 2A8000    LD HL,(SDATA) ;Destination address
E12E 3E01      LD A,1       ;Sector to read is 1
E130 D382      OUT (DSECT),A ;Staticise sector to read

;Ready to boot.....Begin single density boot.
E132 3E0C      BOOT0:    LD A,0CH     ;Configure for SSSD,8,A
E134 D385      OUT (DCONF),A ;Stat FDCI for SSSD,8,A
E136 CD0000    CALL RDSEC   ;Read SSSD,8,A sector 1
E139 CD4BE1    CALL CKSTAT   ;Check for a good read (S/D)

;Single density failed..Try double density
E13C 3E08      LD A,8       ;Set for double density,8,A
E13E D385      OUT (DCONF),A ;Select SSDD,8,A on FDCI
E140 CD0000    CALL RDSEC   ;Read SSDD,8,A sector 1
E143 CD4BE1    CALL CKSTAT   ;Check for a good read (D/D)

;Both modes failed.....Retry sequence 10 times
E146 10EA      DJNZ BOOT0    ;If retries then retry

;Boot failed 10 times...Do a tidy rom stop
E148 C31CE1    JP NBOOT     ;Branch boot not possible

```



```

;Check status, return if error
E14B DB80 CKSTAT: IN A,(DSTAT) ;Get end status
E14D 2A8000 LD HL,(SDATA) ;Data address
E150 E69D AND 9DH ;Mask for errors
E152 324800 LD (RXSTAT),A ;Save status
E155 C0 RET NZ ;Return to retry if error

;Good boot.....Move loader to 0080H
E156 F1 POP AF ;Unstack retry return
E157 118000 LD DE,SDATA ;Destination. HL = source
E15A 018000 LD BC,SDATA ;Size
E15D EDB0 LDIR ;Transfer loader

;Final initialise keyboard again just in case
E15F CD74E4 CALL KINIT

;Loader now at SDATA....Transfer to loader code
E162 C38000 JP SDATA ;Enter loader

;Discom,(sub),gives the command in Acc to the FDCI
;and waits for INTRQ to be returned.
E165 324700 DISCOM: LD (COMMND),A ;Save command
E168 D380 OUT (DCOMM),A ;Send command to FDCI
E16A DB84 QINTRQ: IN A,(DPOLL) ;Poll for flags
E16C E601 AND 1 ;Looking for INTRQ
E16E 28FA JR Z,QINTRQ ;Branch if not done
E170 C9 SPARE1: RET ;Exit command done
E171 C9 SPARE2: RET

;Prnmsg,(sub),prints the string (HL) till (HL) = 0
E172 7E PRNMSG: LD A,(HL) ;Read string character
E173 B7 OR A ;Set up flags on char
E174 C8 RET Z ;Return if zero char
E175 4F LD C,A ;Character to C
E176 CDB6E4 CALL CONOUT ;Print character in C
E179 23 INC HL ;Advance the pointer
E17A 18F6 JR PRNMSG ;Branch for next char

;Compare, (sub), compares HL and DE, returns flags on
;the result.
; Z C S
; HL = DE 1 0 0
; HL < DE 0 1 1
; HL > DE 0 0 0
E17C B7 COMPAR: OR A ;Clear carry
E17D ED52 SBC HL,DE ;Compare HL:DE
E17F 19 ADD HL,DE ;Restore HL value
E180 C9 SPARE3: RET ;Exit flags have compare

;Diskr,(sub),decodes disk error status to the vdu
E181 C5 DISKER: PUSH BC
E182 D5 PUSH DE
E183 E5 PUSH HL
E184 21B9E3 LD HL,SDRIVE ;Drive string to HL
E187 CD72E1 CALL PRNMSG ;Print Drive

```



```

E18A 3A4400      LD A,(DISKNO)    ;Get drive
E18D C641        ADD A,41H        ;Convert to ascii
E18F 4F          LD C,A          ;Drive to print register
E190 CD12E0      CALL BEGIN+12H   ;Print the drive at fault
E193 3A4700      LD A,(COMMND)    ;Command to A
E196 21C3E3      LD HL,SSEEK      ;Seek string to HL
E199 E6E0        AND 0E0H         ;Mask for command type
E19B 2811        JR Z,DERR        ;Jump if seek command
E19D 21C9E3      LD HL,SREAD      ;Read string to HL
E1A0 FE80        CP 80H           ;Was command read?
E1A2 280A        JR Z,DERR        ;Jump if read command
E1A4 21CFE3      LD HL,SWRITE     ;Write string to HL
E1A7 FEA0        CP 0A0H         ;Was command write
E1A9 2803        JR Z,DERR        ;Jump if write command
E1AB 21D6E3      LD HL,SCOMER     ;Unknown command to HL
E1AE CD72E1      DERR: CALL PRNMSG ;Print string in HL
E1B1 21DCE3      LD HL,STRAC      ;Track string to HL
E1B4 CD72E1      CALL PRNMSG      ;Print track string
E1B7 3A4500      LD A,(TRACK)     ;Track to A
E1BA CD14E2      CALL ATODEC      ;Print track number
E1BD 21E4E3      LD HL,SSECT      ;Sector string to HL
E1C0 CD72E1      CALL PRNMSG      ;Print sector string
E1C3 3A4600      LD A,(SECTOR)    ;Sector to A
E1C6 CD14E2      CALL ATODEC      ;Print sector number
E1C9 21EDE3      LD HL,SSTAT      ;Status string to HL
E1CC CD72E1      CALL PRNMSG      ;Print status string
E1CF 3A4800      LD A,(RXSTAT)    ;Status to A
E1D2 2102E4      LD HL,NRDY       ;Not ready to HL
E1D5 17          RLA              ;(7) error "not ready"?
E1D6 382C        JR C,GOERRP      ;Branch if yes
E1D8 210CE4      LD HL,NWRT       ;Write prot to HL
E1DB 17          RLA              ;(6) error "write prot"?
E1DC 3826        JR C,GOERRP      ;Branch if yes
E1DE 17          RLA              ;(5) status left
E1DF 17          RLA              ;(4) error RNF or SKER
E1E0 F5          PUSH AF          ;Stack status (4)
E1E1 3A4700      LD A,(COMMND)    ;Get command code
E1E4 E6E0        AND 0E0H         ;Mask for S=0: R=80: W=A0
E1E6 2007        JR NZ,NSK        ;Jump if not seek command
E1E8 F1          POP AF           ;Restore status seek error
E1E9 214BE4      LD HL,NTRK       ;Seek error to HL
E1EC 3816        JR C,GOERRP      ;Jump if seek error
E1EE F5          PUSH AF          ;Stack status
E1EF F1          NSK: POP AF       ;Restore status
E1F0 211CE4      LD HL,NREC       ;Rec not found to HL
E1F3 380F        JR C,GOERRP      ;Jump if rec not found
E1F5 212DE4      LD HL,NCRC       ;Bad CRC to HL
E1F8 17          RLA              ;(3) error bad CRC
E1F9 3809        JR C,GOERRP      ;Jump if bad CRC
E1FB 2135E4      LD HL,NDAT       ;No data to HL
E1FE 17          RLA              ;(2) error lost data
E1FF 3803        JR C,GOERRP      ;Jump if lost data
E201 213FE4      LD HL,NFRE       ;Busy to HL
E204 CD72E1      GOERRP: CALL PRNMSG ;Print error condition
E207 2156E4      LD HL,STERM      ;Error terminator to HL

```

```

E20A CD72E1      CALL PRNMSG      ;Print error end
E20D E1          POP HL
E20E D1          POP DE
E20F C1          POP BC
E210 3EFF        LD A,0FFH      ;Flag to BDOS for error
E212 B7          OR A           ;Set flags on error
E213 C9          SPARE4: RET     ;Return to BDOS

;Atodec, (sub), displays A in decimal on the vdu
E214 F5          ATODEC: PUSH AF ;Save AF
E215 C5          PUSH BC       ;Save BC
E216 D5          PUSH DE       ;Save DE
E217 E5          PUSH HL       ;Save HL
E218 2139E2      LD HL,PWR10    ;Point to power 10 table
E21B 0600        ATOD1: LD B,0   ;Result = 0
E21D 96          ATOD2: SUB (HL) ;Subtract power 10
E21E FA24E2      JP M,ATOD3     ;Jump if result negative
E221 04          INC B          ;Result = result + 1
E222 18F9        JR ATOD2      ;Loop to try another sub
E224 86          ATOD3: ADD A,(HL);Restore argument
E225 57          LD D,A        ;Argument to D
E226 78          LD A,B        ;Result to A
E227 C630        ADD A,'0'     ;Make result Ascii
E229 4F          LD C,A        ;Ascii result to C reg
E22A CDB6E4      CALL CONOUT    ;Print result
E22D 3E01        LD A,1        ;last power of ten
E22F BE          CP (HL)       ;Test for last used
E230 23          INC HL        ;Pointing to next power
E231 7A          LD A,D        ;Restore argument
E232 20E7        JR NZ,ATOD1   ;Branch if not finished
E234 E1          POP HL        ;End so restore HL
E235 D1          POP DE        ;Restore DE
E236 C1          POP BC        ;Restore BC
E237 F1          POP AF        ;Restore AF
E238 C9          SPARE5: RET    ;Exit done
E239 640A01      PWR10: DEFB 100,10,1 ;Power of 10 table

;Hexhl, (sub), Prints HL as four hex digits on the vdu.
E23C F5          HEXHL: PUSH AF
E23D 7C          LD A,H
E23E CD47E2      CALL HEXACC
E241 7D          LD A,L
E242 CD47E2      CALL HEXACC
E245 F1          POP AF
E246 C9          RET

;Hexacc, (sub), Print Acc as two hex digits to the vdu.
E247 C5          HEXACC: PUSH BC
E248 F5          PUSH AF
E249 1F          RRA
E24A 1F          RRA
E24B 1F          RRA
E24C 1F          RRA
E24D CD71E2      CALL HEXAL
E250 4F          LD C,A

```

```

E251 CDB6E4      CALL CONOUT
E254 F1          POP AF
E255 F5          PUSH AF
E256 CD71E2      CALL HEXAL
E259 4F          LD C,A
E25A CDB6E4      CALL CONOUT
E25D F1          POP AF
E25E C1          POP BC
E25F C9          RET

```

```

E260 2159E4      INITRD: LD HL,TAB1
E263 110000      LD DE,0
E266 0610        LD B,16
E268 3E3E        LD A,3EH
E26A 12          RD1:  LD (DE),A
E26B 86          ADD A,(HL)
E26C 23          INC HL
E26D 13          INC DE
E26E 10FA        DJNZ RD1
E270 C9          RET

```

;Hexal, (sub), Converts low nibble of ACC to ASCII hex in Acc.

```

E271 E60F      HEXAL: AND 0FH
E273 FE0A      CP 0AH
E275 3F        CCF
E276 CE30      ADC A,30H
E278 27        DAA
E279 C9        RET

```

;String space. Contains Ascii messages for printing

```

E27A 436F6E74 SMEM:  DEFB 'Contiguous store 0000-'

```

```

6967756F

```

```

75732073

```

```

746F7265

```

```

20303030

```

```

302D

```

```

E290 00          DEFB 0

```

```

E291 4D616B65 SCHKA  DEFB 'Make drive A "READY".'

```

```

20647269

```

```

76652041

```

```

20225245

```

```

41445922

```

```

2E

```

```

E2A6 00          DEFB 0

```

```

E2A7 0D0A      NSTORE: DEFB CR,LF

```

```

E2A9 20202020      DEFB '*** ERROR ***'

```

```

20202020

```

```

2A2A2A20

```

```

4552524F

```

```

52202A2A

```

```

2A

```

```

E2BE 0D0A      DEFB CR,LF

```

```

E2C0 496E7375      DEFB 'Insufficient RAM. Load halted.'

```

```

66666963

```

```

69656E74

```

```

2052414D
2E20204C
6F616420
68616C74
65642E
E2DF 0D0A          DEFB CR,LF
E2E1 496E7374      DEFM 'Install RAM to equal system size and retry.'
        616C6C20
        52414D20
        746F2065
        7175616C
        20737973
        74656D20
        73697A65
        20616E64
        20726574
        72792E
E30C 0D0A00        DEFB CR,LF,0
E30F 0D0A          SWBER: DEFB CR,LF
E311 5761726D      DEFM 'Warm load failed.'
        206C6F61
        64206661
        696C6564
        2E
E322 00            DEFB 0
E323 0D0A          SLDER: DEFB CR,LF
E325 43502F4D      DEFM 'CP/M - bad load!'
        202D2062
        6164206C
        6F616421
E335 00            DEFB 0
E336 0D0A          STRAP: DEFB CR,LF
E338 48697420      DEFM 'Hit address 0038H - trap stop.'
        61646472
        65737320
        30303338
        48202D20
        74726170
        2073746F
        702E
E356 0D0A          DEFB CR,LF
E358 54686520      DEFM 'The system has crashed. Please reboot.'
        73797374
        656D2068
        61732063
        72617368
        65642E20
        20506C65
        61736520
        7265626F
        6F742E
E37F 00            DEFB 0
E380 0D0A          SCBER: DEFB CR,LF
E382 426F6F74      DEFM 'Boot failed.'
        20666169

```

```

6C65642E
E38E 00 DEF B 0
E38F 0D0A SRETRY: DEF B CR,LF
E391 506C6561 DEF M 'Please press carriage return '
73652070
72657373
20636172
72696765
20726574
75726E20
E3AD 746F2072 DEF M 'to re-boot.'
652D626F
6F742E
E388 00 DEF B 0
E389 0D0A0A SDRIVE: DEF B CR,LF,LF ;Drive
E3BC 44726976 DEF M 'Drive '
6520
E3C2 00 DEF B 0
E3C3 20736565 SSEEK: DEF M 'seek'
6B
E3C8 00 DEF B 0
E3C9 20726561 SREAD: DEF M 'read'
64
E3CE 00 DEF B 0
E3CF 20777269 SWRITE: DEF M 'write'
7465
E3D5 00 DEF B 0
E3D6 203F3F3F SCOMER: DEF M ' ????'
3F
E3DB 00 DEF B 0
E3DC 20747261 STRAC: DEF M 'track '
636B20
E3E3 00 DEF B 0
E3E4 20736563 SSECT: DEF M 'sector '
746F7220
E3EC 00 DEF B 0
E3ED 0D0A SSTAT: DEF B CR,LF
E3EF 52656365 DEF M 'Received status - '
69766564
20737461
74757320
2D20
E401 00 DEF B 0
E402 4E6F7420 NRDY: DEF M 'Not ready'
72656164
79
E40B 00 DEF B 0
E40C 57726974 NWRT: DEF M 'Write protected'
65207072
6F746563
746564
E41B 00 DEF B 0
E41C 5265636F NREC: DEF M 'Record not found'
7264206E
6F742066

```

```

        6F756E64
E42C 00          DEFB 0
E42D 42616420 NCRC:  DEFM 'Bad CRC'
        435243
E434 00          DEFB 0
E435 4C6F7374 NDAT:  DEFM 'Lost data'
        20646174
        61
E43E 00          DEFB 0
E43F 44726976 NFRE:  DEFM 'Drive busy?'
        65206275
        73793F
E44A 00          DEFB 0
E44B 5365656B NTRK:  DEFM 'Seek error'
        20657272
        6F72
E455 00          DEFB 0
E456 0D0A00 STERM:  DEFB CR,LF,0 ;Error sequence end
E459 4E47AD6D TAB1:  DEFB 4EH,47H,0ADH,6DH ;Sector reader table
E45D 63D8D4DF      DEFB 63H,0D8H,0D4H,0DFH
E461 A8F4ACD7      DEFB 0A8H,0F4H,0ACH,0D7H
E465 0AFCC900      DEFB 0AH,0FCH,0C9H,00H

;Listst,(sub),returns printer status. Busy/not busy.
E469 3EFF      LISTST: LD A,0FFH ;Always not busy
E46B B7        OR A ;Stat flags
E46C C9        RET ;Exit not busy

;Reader,(sub),is not used yet.
E46D 3E1A      READER: LD A,1AH ;EOF for now
E46F E67F      AND 7FH ;No action yet
E471 C9        RET ;Exit nothing done

;Punch,(sub),is not used yet.
E472 79        PUNCH: LD A,C ;No action yet
E473 C9        RET ;Exit nothing done

;Keyboard initilisation. RET if not needed.
E474 AF        KINIT: XOR A ;Clear Acc
E475 324200      LD (KEYCDE),A ;Clear keystore
E478 DB40        IN A,(40H) ;Reset key latch
E47A C9        RET ;Exit done

;Video initilisation routine.
;CLS VDU 2K See COLDST also
E47B 2100F0      CVDU: LD HL,VDU
E47E 1101F0      LD DE,VDU+1
E481 010006      LD BC,24*64
E484 3620        LD (HL),20H
E486 EDB0        LDIR
E488 C9        RET

;LK1 KEYBOARD DRIVER CODE IS 01
;Const,(sub), returns valid key status in Acc
E489 3A4200      CONST: LD A,(KEYCDE) ;Read keycode store

```



```

E48C B7          OR A          ;Set flags on keycode
E48D 200E        JR NZ,OLDKEY  ;Branch with old key
E48F DB40        NEWKEY: IN A,(KDATA) ;Read keyboard
E491 C680        ADD A,80H      ;Strobe to carry flag
E493 324200      LD (KEYCODE),A ;Save keycode
E496 3805        JR C,OLDKEY    ;If strobe go with key
E498 AF          XOR A          ;Clear Acc
E499 324200      LD (KEYCODE),A ;Clear keycode store
E49C C9          RET           ;Exit no key
E49D 3EFF        OLDKEY: LD A,0FFH ;Flag for key
E49F C9          RET           ;Exit with key

;Conin,(sub),returns a valid key from the keyboard
E4A0 CD89E4      CON1: CALL CONST ;Look for a key
E4A3 3A4200      CONIN: LD A,(KEYCODE) ;Get result
E4A6 B7          OR A          ;Set flags on key
E4A7 28F7        JR Z,CON1      ;Branch if no key
E4A9 4F          LD C,A         ;Key to C temp
E4AA DB40        STB: IN A,(KDATA) ;Read for strobe
E4AC C680        ADD A,80H      ;Strobe to carry
E4AE 38FA        JR C,STB       ;Branch if strobe
E4B0 AF          XOR A          ;Clear A
E4B1 324200      LD (KEYCODE),A ;Clear keycode
E4B4 79          LD A,C         ;Key back to A
E4B5 C9          RET           ;Exit key in A

;VDU 2K DRIVER.
;Conout,(sub),sends the C reg to the VDU.
E4B6 C5          CONOUT: PUSH BC ;Save BC
E4B7 D5          PUSH DE       ;Save DE
E4B8 E5          PUSH HL       ;Save HL
E4B9 2A4000      LD HL,(CURSOR) ;HL = cursor
E4BC 3A4300      LD A,(CURCAR) ;Character under cursor
E4BF 77          LD (HL),A     ;Put cursor character back
E4C0 79          LD A,C        ;Get character to display
E4C1 CBBF        RES 7,A       ;Reset parity bit
E4C3 FE0D        CP CR         ;Is it carriage return?
E4C5 200E        JR NZ,NCR     ;Branch if not CR
E4C7 21C0F5      LD HL,LINE24 ;Point to bottom line
E4CA 7E          LD A,(HL)     ;Get this character in A
E4CB 324300      LD (CURCAR),A ;Save A as curcar
E4CE 224000      EXIT: LD (CURSOR),HL ;Save cursor position
E4D1 E1          POP HL        ;Restore HL
E4D2 D1          POP DE        ;Restore DE
E4D3 C1          POP BC        ;Restore BC
E4D4 C9          RET           ;Exit done
E4D5 FE0A        NCR: CP LF     ;Is it line feed?
E4D7 2021        JR NZ,NLF     ;Branch if not LF
E4D9 1100F0      LFEEED: LD DE,VDU ;Point to screen top
E4DC 2140F0      LD HL,VDU+COLS ;Point to line 2 start
E4DF 01C005      LD BC,(LINES-1)*COLS ;Number to move
E4E2 EDB0        LDIR          ;Scroll up
E4E4 21C0F5      LD HL,LINE24 ;Load bottom line
E4E7 0640        LD B,COLS      ;Load line length
E4E9 3E20        LD A,SPACE     ;Space code

```

```

E4EB 77      CL24: LD (HL),A      ;Blank first
E4EC 23      INC HL              ;Advance pointer
E4ED 10FC    DJNZ CL24          ;Blank bottom line
E4EF 21C0F5  LD HL,LINE24       ;Load bottom line
E4F2 7E      SAVCUR: LD A,(HL)  ;Get cursor character
E4F3 324300  LD (CURCAR),A      ;Save it in curcar
E4F6 365F    LD (HL),'_'       ;Print the cursor
E4F8 18D4    JR EXIT            ;Branch to exit
E4FA FE08    NLF: CP BS         ;Is it backspace?
E4FC 2005    JR NZ,NBS         ;Branch if not
E4FE 3620    LD (HL),SPACE     ;Blot out character
E500 2B      DEC HL            ;Backspace pointer
E501 18EF    JR SAVCUR         ;Branch to exit
E503 77      NBS: LD (HL),A     ;Print code in A
E504 23      INC HL            ;Advance the cursor
E505 1100F6  LD DE,LINE24+COLS ;Screen end
E508 CD7CE1  CALL COMPAR       ;Does cursor = end?
E50B 28CC    JR Z,LFEED        ;Branch if end
E50D 18E3    JR SAVCUR         ;Branch if not end

;PRINTER DRIVER (LISTST is standard, not busy).
;Standard printer driver.
;Port 6 is status, bit 7 is TBMT
;Port 7 is data.
;List,(sub),prints the A reg on the printer.
E50F F5      LIST: PUSH AF      ;Save character to print
E510 DB06    LIST1: IN A,(6)    ;Get printer status.P2
E512 E680    AND 80H           ;Is buffer empty?
E514 28FA    JR Z,LIST1        ;Loop if no.
E516 F1      POP AF           ;Get character to print
E517 D307    OUT (7),A         ;Pass it to the printer.P3
E519 C9      RET              ;Exit done.

E51A 0D0A    SGD: DEFB CR,LF
E51C 494E5445 DEFM 'INTERAK DOS V1'
      52414B20
      444F5320
      5631
E52A 0D0A    DEFB CR,LF
E52C 424F4F54 DEFM 'BOOT 0301-0100-018401.'
      20303330
      312D3031
      30302D30
      31383430
      312E
E542 0D0A00  DEFB CR,LF,0
      END

ERRORS=0000

```

## LETTERS TO THE EDITOR

Please write with comments, ideas, complaints and suggestions. Name and address must be enclosed. Responsibility for views and comments expressed cannot be held by the editor as members letters are published with the minimum changes (deleted bad language etc). (Note; I type what I see, if you forget a word then it will be missing in the newsletter, also if you spell a word wrong then it's quite likely that I will punch it in wrong.)

Dear Ed,

Yes, it would be a good idea to extend the Interak news letter to cover Custom 80 users as well. There are not enough of us for a separate user group. I enclose 2 contributions: a constuctional note and a demo program showing how the CTC on the cassette board might be used for multi-tasking.

I am working on the artwork for some new circuit boards which I hope to get produced shortly; these are : a tracked version of Mike Hastings Sound Generator Board, a Controller Board for electronic musical instruments (synthesizers ect), and a PIO board. Further off are an RS232 board (send and recive) and a sound sampling board (fast D -> A and A -> D). These boards will not be up to current Greenbank standards of production etc but they should work and be usefull.

DON COX, 13 OXFORD RD, MIDDLESBROUGH.

Dear Ed,

## POINT-CONTACT TAPES

Would'ent you like to have a chat now and again with other Interakers? seek help with that latest project or can you offer help with adding joysticks! or what ever it may be, etc.

Problem there's not another Interaker around the corner just a chap with a Spectrum or Vic-20.

Well this is what you do, join P.C.Tapes, small groups of folks usually 3 spend about 20 minutes talking to each other on a standard C60 (or 4 on a C90) cassette tape.

Groups can be formed to cover special topics, you also get to know some very interesting people from different parts of the country. The only cost is the postage sending it on to the next member. No hard rules or regulations except it would be unfair to keep the tape for too long, and other members waiting!

You don't have to talk computers, how about Photography or may be Gardening, etc.....Interested drop me a line!!!

MEL SAUNDERS, 7 DRUMCLIFF ROAD, THURNBY LODGE, LEICESTER, LE5 2LH.

Dear Ed,

If the sketch (below) is what the Zybasic Manual means by a 'three dimensioned array', the program enclosed will produce the correct values for the 'P' and 'N' indices in the plane at 'D'.

But maybe this is not what the manual asks for? (I haven't got the hang of STR\$ yet, and I thought "arrays" was something to do with flower arranging!!)

Add this on to the prog on page 4-20 (Zybasic Manual)

```

1000 X=5: V=4           X=Horizontal (length)
1020 INPUT N: INPUT P: INPUT D   V=Vertical (hight)
1021 S=V*X               S calculates one full plane
1025 IF D=0 S=0           If only 2 dim array required
1030 PRINT (N*X+P-X)+S*D   Gives value at 'D' depth of position
                          .."seen" on 2 dim array

1035 PRINT: PRINT
1040 GOTO 1000           Saves me having to type G.1000
                          (...and getting it wrong!)

```

```

<----- D ----->
^ D0          D1          D2          D3
! !\          !\          !\          !\
! ! !\        !\ !\        !\ !\        !\ !\
! ! 1 !\      !17 !\      !33 !\      !49 !\
! !\ !\      !\ !\ !\      !\ !\ !\      !\ !\ !\
! ! !\ 2 !\   !\ !18 !\   !\ !34 !\   !\ !50 !\
! ! 5 !\ !\   !21 !\ !\   !37 !\ !\   !53 !\ !\
N !\ !\ !\ 3 !\ !\ !\ 19 !\ !\ !\ 35 !\ !\ !\ 51 !\
! ! !\ 6 !\ !\ !\ 22 !\ !\ !\ 38 !\ !\ !\ 54 !\ !\
! ! 9 !\ !\ !\ 4 !\ 25 !\ !\ 20 !\ 41 !\ !\ 36 !\ 57 !\ !\ 52 !\
! !\ !\ !\ 7 !\ !\ !\ 23 !\ !\ !\ 39 !\ !\ !\ 55 !\ !\
! ! !\ 10 !\ !\ !\ 26 !\ !\ !\ 42 !\ !\ !\ 58 !\ !\
! !13 !\ !\ !\ 8 !\ 29 !\ !\ 24 !\ 45 !\ !\ 40 !\ 61 !\ !\ 56 !\
V ! ! !\ 11 !\ !\ !\ 27 !\ !\ !\ 43 !\ !\ !\ 59 !\ !\
! ! !\ 14 !\ !\ !\ 30 !\ !\ !\ 46 !\ !\ !\ 62 !\ !\
! !\ !\ !\ 12 !\ !\ !\ 28 !\ !\ !\ 44 !\ !\ !\ 60 !\ !\
! ! !\ 15 !\ !\ !\ 31 !\ !\ !\ 47 !\ !\ !\ 63 !\ !\
! !\ !\ !\ 16 !\ !\ !\ 32 !\ !\ !\ 48 !\ !\ !\ 64 !\ !\
X ! ! !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\
P ! ! !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\
! ! !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\ !\

```

P.S Tried it with 8\*9\*6 and it still appears to work.

P.P.S. Now I suppose the Newsletter will want us to write a prog to solve Rubics problem!.

DAVE BALLAM, 13 WOOLIFERS AVE, CORRINGHAM, ESSEX, SS17 9AU.

Dear ED,

I was very excited by issue 6 of IUGN, in particular the article on modems. I am taking you up on your offer and I'm going to talk a little about my latest project for my Interak:

For a few months I have been trying to connect my Interak to the telephone network, so I have researched the subject in some detail and here are some ideas on the subject.

#### HARDWARE

Anything connected to the B.T. telephone network must be B.T. approved. So by far the easiest way to obtain such approved equipment is to buy it. I purchased some time ago an old Prestel adapter which will do just nicely. Since then however I have learned of a more suitable MODEM. For about £50 (+vat) a UNICOM MODEM can be purchased which offers the following features:

Full and Half Duplex, Auto Dial, and Auto Baud rate sensor and, if that wasn't enough originate and answer modes! I wish I had seen this before I acquired my Prestel adaptor!!

The Unicom is available at the moment for the BBC, but is soon to be available for the Spectrum, but I suspect it would be easy to interface to Interak, as it may accept an RS232 link. Unfortunately I do not know where to get them from, so keep your eyes peeled for an advertisement.

If such a device is utilised interfacing should be quite easy. The software to drive it could be adapted, or re-written, from the Spectrum, which is also Z80. (I also have a Spectrum!). With such a set up users could access other networks, eg Prestel, Micronet, as well as our own if we set one up, and other users directly.

The other hardware required would be a modern B.T. socket, at a cost of £27.00p I think, if not already fitted, and the interface hardware. As I have already said this may only be an RS232 port, (or some other serial communication device) and perhaps a parallel port to do some switching. (£30?).

#### SOFTWARE

The software for various networks varies, so if Interak were to create it's own network then we may as well create our own standards. We would need software to 1. Control the modem, (machine code routines to handle dialing, receiving and sending data), 2. For error checking and correction, (errors occurring due to noise on the telephone network), 3. Protocol. I have said protocol because this can take many forms which may be quite complex to deal with.

#### PROTOCOL

Such a modem link between two computers must be ultimately controlled by protocol, a common language between the two devices, such a language must be able to cope with all that the user required, eg. The access to centralised data, transfer of programs, graphic displays, menus, and other uses of the systems.

I do not want to go on, but such a protocol system could enable you to enter the details of what you want from the main system, then the user computer calls the main system, the access and transfer is made and the link broken. Other systems may not allow you to request data while the link is made. This will be expensive in phone calls!

I have a more detailed plan of a protocol system which would work on Interak. If interest is high I can write again if you require, and I could give a more detailed explanation of data networks.

STEVE BRUMBY, 89 ST MARGARETS ROAD, BRADFORD, W.YORKS, BD7 2BY.



PRINT DECIMAL OF HL REGISTER PAIR  
BY R.ELDRIDGE.

;Print HL register as a decimal number  
;Taken from an algorithm that appears in :-  
;"Seminumerical Algorithms" Knuth. Addison-Wesley 1969.  
;Although it is my byline I only translated this routine  
;from code written for the 8080 onto my Z80. Ah well! no  
;fame in honesty.  
;AF is destroyed

ORG 0

```

0000 C5      HL2DEC: PUSH BC      ;Save BC
0001 D5      PUSH DE      ;Save DE
0002 E5      PUSH HL      ;Save HL
0003 01F6FF  LD BC,-10      ;Set up for repeated subtraction by 10
0006 11FFFF  LD DE,-1      ;Quotient of result
0009 09      HL2D1: ADD HL,BC   ;Remainder = Remainder - 10
000A 13      INC DE         ;Quotient = Quotient + 1
000B DA0900  JP C,HL2D1      ;Flag C until HL goes negative then NC
000E 010A00  LD BC,10      ;Ready to add back the underflow 10
0011 09      ADD HL,BC      ;Remainder = Remainder + 10
0012 EB      EX DE,HL       ;DE has remainder, HL has quotient
0013 7C      LD A,H         ;Check if Quotient zero
0014 B5      OR L
0015 C40000  CALL NZ,HL2DEC     ;If Quotient non zero loop
0018 7B      HL2D2: LD A,E     ;Get digit from stacked result
0019 C630    ADD A,'0'        ;Convert digit to ASCII
001B 4F      LD C,A          ;Move result digit to C for ...
001C CD0000  CALL CONOUT      ;... conout routine (see UGN-7 p19)
001F E1      POP HL          ;Restore HL
0020 D1      POP DE          ;Restore DE
0021 C1      POP BC          ;Restore BC
0022 C9      RET             ;Either to HL2D2 or exit done.

0000        CONOUT EQU 0     ;Put in where yours is

```

END

```

CONOUT 0000      ;Put in a pointer to your Conout routine.
HL2D1  0009
HL2D2  0018
HL2DEC 0000

```



## INTERAKTION BOOK LIBRARY

This section is to give members access to a wide range of books on computing and electronics. The only cost to the member is that of postage. Books may be borrowed for up to 4 weeks. At present the books available are :-

## LANGUAGE BOOKS

TRS 80 Assembly Language Programming ... Radio Shack  
 Z80 Assembly Language Programming Manual. Zilog  
 A Course in Basic Programming ..... Sinclair  
 Making the Most of your ZX 80 ..... Tim Hartnell  
 30 Hour Basic ..... C. Prigmore  
 Basic for Home Computers ..... B. Albrecht, L. Finkel & J. Brown  
 Course in Standard Coral 66 ..... J. D. Halliwell & T. A. Edwards  
 Simple Pascal ..... J. McGregor & A. Watt  
 Lecture Notes in Computer Science  
 Pascal User Manual and Report ..... K. Jensen & N. Wirth

## DATA BOOKS

Mostek 1982/1983 Microelectronic Data Book (mem/CPU/Per) .. Mostek  
 Memory Data Book & Designers Guide 1980 ..... Mostek  
 Bytewyde Memory Data Book 1981 ..... Mostek  
 National Semiconductor Memory Data Book (1980) ..... National  
 National Semiconductor Interface Data Book (1980) ..... National  
 TTL Data Book ..... National  
 The European Selection (mem/inface/lin) ..... Motorola

## GENERAL &amp; ELECTRONICS

Computer Technology for Technicians  
 and Technical Engineers Vol. 1 ..... R. Watkin  
 Electronic Computers Made Simple ..... H. Jacobowitz  
 Test Instruments for Electronics  
 (how to build test instruments) ..... M. Clifford  
 Practical Test Instruments You Can Build ... W. Green  
 How to Troubleshoot & Repair  
 Electronic Test Equipment ..... M. Horowitz  
 Computers and the Social Sciences ..... A. Brier & I. Robinson

## MANUALS etc.

Epson MX-80 Type II Operation Manual ..... Epson  
 Newbury 8000 Series VDU Terminal  
 Operator Instruction Manual ..... Newbury Labs  
 Electronics Projects Index ..... Polytechnic  
 Why Do You Need a Personal Computer? ..... Leventhal & Straffars  
 Computer Programming in the Classroom ..... B. J. Jackson  
 TABS Accounting Business Systems User Guide Vol 1. TABS  
 Easy Add-on Projects, Spectrum, ZX-81, Jupiter Ace .. Owen Bishop  
 6502 Games ..... Rodney Zaks

All books have been donated by users. If you have any books etc. surplus to requirements please let me have them.

See Contacts page at the end of this issue for "BOOKS" address.

Richard Bowyer

## SOFTWARE LIBRARY

See Contacts page at the end of this issue for "ORDER FROM" addresses. Software supplied is the responsibility of the "ORDER FROM". Please deal directly with the "ORDER FROM" in the event of bugs ect. You may use this section to sell software to other users. Send a single line entry, as below, to the EDITOR. Note that you will be responsible for the distribution and support of your own product. The newsletter cannot be held responsible for or get involved with duff code or distributors. Of course we will publish letters deriding any product that fails to live up to its claims.

NAME	DESCRIPTION	CODE	VDU	ORDER FROM	COST
INTERAK BULLETIN BOARD SOFTWARE		?	?	D.MASON	£ 4.00
ZYMON 2.003	INTERAK monitor	MC	A	GREENBANK	£15.95
ZYMON 2.203	INTERAK monitor	MC	C	GREENBANK	£15.95
ZYBASIC 2A	INTERAK BASIC (tape)	MC	A	GREENBANK	£14.95
ZYBASIC 2C	INTERAK BASIC (Rom)	MC	A	GREENBANK	£27.75
ZYBASIC 3A	INTERAK BASIC (tape)	MC	C	GREENBANK	£15.95
ZYBASIC 3C	INTERAK BASIC (Rom)	MC	C	GREENBANK	£27.75
XTAL BASIC	14K BASIC	MC	A	USER GROUP	£40.00
XTAL BASIC	14K BASIC	MC	C	USER GROUP	£40.00
FIGFORTH	Forth Compiler	MC	A	USER GROUP	£15.00
FIGFORTH	Forth Compiler	MC	C	USER GROUP	£15.00
ASM 32	Editor Assembler	MC	A	USER GROUP	£10.00
ASM 64	Editor Assembler	MC	C	USER GROUP	£10.00
HC DISASS	Simple Disassembler	MC	A	USER GROUP	£ 3.00
REVAS	Better Disassembler	MC	A	USER GROUP	POA
MEGABUG	Debug/Training Package	MC	C	USER GROUP	£13.00
VELTEXT	Text Editor	MC	A	USER GROUP	£ 5.00
VELTEXT	Text Editor	MC	C	USER GROUP	£ 5.00
Lander	Lander Game	XL	C	USER GROUP	PP
Towers	Towers Puzzle	XL	C	USER GROUP	PP
Crazy Maze	"3D" Maze Game	XL	C	USER GROUP	PP
Avalanche	Blob Dodging Game	ZB2	A	USER GROUP	PP
Monster Mash	Maze Game	ZB2	A	USER GROUP	PP
Graph	Graph Plotter	ZB2	A	USER GROUP	PP
Rakovsky	Computer Chess (6 levs)	MC	A	USER GROUP	\
AC10.XX	(Chess men EPROM VDU-K)		A	USER GROUP	/ £ 3.00
Rakovsky	Computer Chess (6 levs)	MC	C	USER GROUP	\
AC10.XX	(Chess men EPROM VDU-2K)		C	USER GROUP	/ £ 3.00
Happy Sums	Fun maths	ZB2	A	USER GROUP	PP
Hangman	Spelling game	ZB2	A	USER GROUP	PP
0's and X's	Game	ZB2	A	USER GROUP	PP
Pools Pick	Random Draw Selector	ZB2	A	USER GROUP	PP
Count	Learn to count	ZB2	A	USER GROUP	PP
Dice Pontoon	Simple Game	ZB2	A	USER GROUP	PP

Key: MC = machine code.      Screen: A    32 x 24 VDU-K  
 ZB2 = ZYBASIC.                    B    64 x 16 VDU-1K  
 XL = XTAL BASIC.                    C    64 x 24 VDU-2K  
 PP = Postage & Packing.  
 POA = Please enquire (Price on Application).

# CONTACTS

- BACK ISSUES... To obtain Back Issues write to :-  
Mr D.Parkins, Greenbank Electronics, 92 New Chester road,  
New Ferry, Wirral, Merseyside, L62 5AG.
- BOOKS..... To borrow or supply books contact :-  
Mr R.E.Bowyer, 45 Ford drive, Yarnfield, Stone, Staffs.
- BULLETIN BOARD. For information and software contact :-  
Mr D.Mason, Pererill Court, Mooragh Promenade, Ramsey,  
Isle of man.
- D.MASON..... Mr D.Mason, Pererill Court, Mooragh Promenade, Ramsey,  
Isle of man.
- EDITOR..... Send newsletter contributions to :-  
Mr R.Eldridge, 28 Wycherley Close, Blackheath, London,  
SE3 7QH. Please no phone calls.
- GREENBANK .... To obtain software from Greenbank write to :-  
Greenbank Electronics Ltd, 92 New Chester road,  
New Ferry, Wirral, Merseyside, L62 5AG.
- MEMBERSHIP.... For membership information contact :-  
Mr P.P.Vella, 19 Ford drive, Yarnfield, Staffs.
- P.C.TAPES..... For details contact :-  
Mr M.Saunders, 7 Drumcliff road, Thurnby Lodge,  
Leicester, LE5 2LH.
- SUBSCRIPTIONS. For IUG subscriptions contact :-  
Mr P.P.Vella, 19 Ford drive, Yarnfield, Staffs.
- USER GROUP ... To obtain software from the User Group write to :-  
Mr P.P.Vella, 19 Ford Drive, Yarnfield, Staffs.
-